A Statistical Model for Multilingual Entity Detection and Tracking

R. Florian, H. Hassan*, **A. Ittycheriah, H. Jing**
N. Kambhatla, X. Luo, N. Nicolov, and **S. Roukos**
I.B.M. T.J. Watson Research Center
Yorktown Heights, NY 10598
{raduf,abei,hjing,nanda,xiaoluo, nicolas,roukos}@us.ibm.com
*hanyh@eg.ibm.com

Abstract

Entity detection and tracking is a relatively new addition to the repertoire of natural language tasks. In this paper, we present a statistical language-independent framework for identifying and tracking named, nominal and pronominal references to entities within unrestricted text documents, and chaining them into clusters corresponding to each logical entity present in the text. Both the mention detection model and the novel entity tracking model can use arbitrary feature types, being able to integrate a wide array of lexical, syntactic and semantic features. In addition, the mention detection model crucially uses feature streams derived from different named entity classifiers. The proposed framework is evaluated with several experiments run in Arabic, Chinese and English texts; a system based on the approach described here and submitted to the latest Automatic Content Extraction (ACE) evaluation achieved top-tier results in all three evaluation languages.

1 Introduction

Detecting entities, whether named, nominal or pronominal, in unrestricted text is a crucial step toward understanding the text, as it identifies the important conceptual objects in a discourse. It is also a necessary step for identifying the relations present in the text and populating a knowledge database. This task has applications in information extraction and summarization, information retrieval (one can get all hits for Washington/person and not the ones for Washington/state or Washington/city), data mining and question answering.

The Entity Detection and Tracking task (EDT henceforth) has close ties to the *named entity recognition* (NER) and *coreference resolution* tasks, which have been the focus of attention of much investigation in the recent past (Bikel et al., 1997; Borthwick et al., 1998; Mikheev et al., 1999; Miller et al., 1998; Aberdeen et al., 1995; Ng and Cardie, 2002; Soon et al., 2001), and have been at the center of several evaluations: MUC-6, MUC-7, CoNLL'02 and CoNLL'03 shared tasks. Usually, in computational linguistic literature, a named entity represents an instance of a name, either a location, a person, an organization, and the NER task consists of identifying each individual occurrence of such an entity. We will instead adopt the nomenclature of the Automatic Content Extraction program[1] (NIST, 2003a): we will call the instances of textual references to objects or abstractions *mentions*, which can be either named (e.g. *John Mayor*), nominal (e.g. *the president*) or pronominal (e.g. *she, it*). An *entity* consists of all the mentions (of any level) which refer to one conceptual entity. For instance, in the sentence

President _John Smith_ said _he_ has no comments.

there are two mentions: *John Smith* and *he* (in the order of appearance, their levels are named and pronominal), but one entity, formed by the set {*John Smith, he*}.

In this paper, we present a general statistical framework for entity detection and tracking in unrestricted text. The framework is not language specific, as proved by applying it to three radically different languages: Arabic, Chinese and English. We separate the EDT task into a mention detection part – the task of finding all mentions in the text – and an entity tracking part – the task of combining the detected mentions into groups of references to the same object.

The work presented here is motivated by the ACE evaluation framework, which has the more general goal of building multilingual systems which detect not only entities, but also *relations* among them and, more recently, *events* in which they participate. The EDT task is arguably harder than traditional named entity recognition, because of the additional complexity involved in extracting non-named mentions (nominals and pronouns) and the requirement of grouping mentions into entities.

We present and evaluate empirically statistical models for both mention detection and entity tracking problems. For mention detection we use approaches based on Maximum Entropy (MaxEnt henceforth) (Berger et al., 1996) and Robust Risk Minimization (RRM henceforth)

[1] For a description of the ACE program see `http://www.nist.gov/speech/tests/ace/`.

(Zhang et al., 2002). The task is transformed into a sequence classification problem. We investigate a wide array of lexical, syntactic and semantic features to perform the mention detection and classification task including, for all three languages, features based on pre-existing statistical semantic taggers, even though these taggers have been trained on different corpora and use different semantic categories. Moreover, the presented approach implicitly learns the correlation between these different semantic types and the desired output types.

We propose a novel MaxEnt-based model for predicting whether a mention should or should not be linked to an existing entity, and show how this model can be used to build entity chains. The effectiveness of the approach is tested by applying it on data from the above mentioned languages — Arabic, Chinese, English.

The framework presented in this paper is language-universal – the classification method does not make any assumption about the type of input. Most of the feature *types* are shared across the languages, but there are a small number of useful feature types which are language-specific, especially for the mention detection task.

The paper is organized as follows: Section 2 describes the algorithms and feature types used for mention detection. Section 3 presents our approach to entity tracking. Section 4 describes the experimental framework and the systems' results for Arabic, Chinese and English on the data from the latest ACE evaluation (September 2003), an investigation of the effect of using different feature types, as well as a discussion of the results.

2 Mention Detection

The mention detection system identifies the named, nominal and pronominal mentions introduced in the previous section. Similarly to classical NLP tasks such as base noun phrase chunking (Ramshaw and Marcus, 1994), text chunking (Ramshaw and Marcus, 1995) or named entity recognition (Tjong Kim Sang, 2002), we formulate the mention detection problem as a classification problem, by assigning to each token in the text a label, indicating whether it starts a specific mention, is inside a specific mention, or is outside any mentions.

2.1 The Statistical Classifiers

Good performance in many natural language processing tasks, such as part-of-speech tagging, shallow parsing and named entity recognition, has been shown to depend heavily on integrating many sources of information (Zhang et al., 2002; Jing et al., 2003; Ittycheriah et al., 2003). Given the stated focus of integrating many feature types, we are interested in algorithms that can easily integrate and make effective use of diverse input types. We selected two methods which satisfy these criteria: a linear classifier – the Robust Risk Minimization classifier – and a log-linear classifier – the Maximum Entropy classifier. Both methods can integrate arbitrary types of information and make a classification decision by aggregating all information available for a given classification.

Before formally describing the methods[2], we introduce some notations: let $\mathcal{C} = \{c_1, \ldots, c_n\}$ be the set of predicted classes, $\mathcal{X}$ be the example space and $\mathcal{F} = \{0,1\}^m$ be the feature space. Each example $x \in \mathcal{X}$ has associated a vector of binary features $f(x) = (f_1(x), \ldots, f_m(x))$. We also assume the existence of a training data set $\mathcal{T} \subset \mathcal{X}$ and a test set $\mathcal{E} \subset \mathcal{X}$.

The RRM algorithm (Zhang et al., 2002) constructs n linear classifiers $(C_i)_{i=1\ldots n}$(one for each predicted class), each predicting whether the current example belongs to the class or not. Every such classifier C_i has an associated feature weight vector, $(w_{ij})_{j=1\ldots m}$, which is learned during the training phase so as to minimize the classification error rate[3]. At test time, for each example $x \in \mathcal{E}$, the model computes a score

$$a_i(x) = \sum_{j=1}^{m} w_{ij} \cdot f_j(x)$$

and labels the example with either the class corresponding to the classifier with the highest score, if above 0, or *outside*, otherwise. The full decoding algorithm is presented in Algorithm 1. This algorithm can also be used for *sequence classification* (Williams and Peng, 1990), by converting the activation scores into probabilities (through the *soft-max* function, for instance) and using the standard dynamic programing search algorithm (also known as Viterbi search).

Algorithm 1 The RRM Decoding Algorithm

```
foreach $x \in \mathcal{E}$
  foreach $i = 1 \ldots n$
    $a_i[x] = \sum_{j=1}^{m} w_{ij} \cdot f_j(x)$
  $class(x) \leftarrow \arg\max_i a_i[x]$
```

Somewhat similarly, the MaxEnt algorithm has an associated set of weights $(\alpha_{ij})_{j=1\ldots m}^{i=1\ldots n}$, which are estimated during the training phase so as to maximize the likelihood of the data (Berger et al., 1996). Given these weights, the model computes the probability distribution of a particular example x as follows:

$$P(c_i|x) = \frac{1}{Z} \prod_{j=1}^{m} \alpha_{ij}^{f_j(x)}, \; Z = \sum_i \prod_j \alpha_{ij}^{f_j(x)}$$

where Z is a normalization factor.

After computing the class probability distribution, the assigned class is the most probable one a posteriori. The sketch of applying MaxEnt to the test data is presented in Algorithm 2. Similarly to the RRM model, we use the model to perform sequence classification, through dynamic programing.

[2] This is not meant to be an in-depth introduction to the methods, but a brief overview to familiarize the reader with them.

[3] Actually, the optimizing function contains a regularization factor which considerably improves the robustness of the system — for full details, see Zhang et al. (2002).

Algorithm 2 The MaxEnt Decoding Algorithm

```
foreach $x \in \mathcal{E}$
  $Z \leftarrow 0$
  foreach $i = 1 \ldots n$
    $p_i[x] = \prod_{j=1}^{m} \alpha_{ij}^{f_j(x)}$
  Normalize (p)
  $class(x) \leftarrow \arg\max_i p_i[x]$
```

Within this framework, any type of feature can be used, enabling the system designer to experiment with interesting feature types, rather than worry about specific feature interactions. In contrast, in a rule based system, the system designer would have to consider how, for instance, a WordNet (Miller, 1995) derived information for a particular example interacts with a part-of-speech-based information and chunking information. That is not to say, ultimately, that rule-based systems are in some way inferior to statistical models – they are built using valuable insight which is hard to obtain from a statistical-model-only approach. Instead, we are just suggesting that the output of such a system can be easily integrated into the previously described framework, as one of the input features, most likely leading to improved performance.

2.2 The Combination Hypothesis

In addition to using rich lexical, syntactic, and semantic features, we leveraged several pre-existing mention taggers. These pre-existing taggers were trained on datasets outside of ACE training data and they identify types of mentions different from the ACE types of mentions . For instance, a pre-existing tagger may identify dates or occupation mentions (not used in ACE), among other types. It could also have a class called *PERSON*, but the annotation guideline of what represents a *PERSON* may not match exactly to the notion of the *PERSON* type in ACE. Our hypothesis – the *combination hypothesis* – is that combining pre-existing classifiers from diverse sources will boost performance by injecting complementary information into the mention detection models. Hence, we used the output of these pre-existing taggers and used them as additional feature streams for the mention detection models. This approach allows the system to *automatically* correlate the (different) mention types to the desired output.

2.3 Language-Independent Features

Even if the three languages (Arabic, Chinese and English) are radically different syntacticly, semantically, and even graphically, all models use a few universal types of features, while others are language-specific. Let us note again that, while some types of features only apply to one language, the models have the same basic structure, treating the problem as an abstract classification task.

The following is a list of the features that are shared across languages (w_i is considered by default the current token):

- tokens[4] in a window of 5: $\{w_{i-2}, \ldots, w_{i+2}\}$;
- the part-of-speech associated with token w_i
- dictionary information (whether the current token is part of a large collection of dictionaries - one boolean value for each dictionary)
- the output of named mention detectors trained on different style of entities.
- the previously assigned classification tags[5].

The following sections describe in detail the language-specific features, and Table 1 summarizes the feature types used in building the models in the three languages. Finally, the experiments in Section 4 detail the performance obtained by using selected combinations of feature subsets.

2.4 Arabic Mention Detection

Arabic, a highly inflected language, has linguistic peculiarities that affect any mention detection system. An important aspect that needs to be addressed is segmentation: which style should be used, how to deal with the inherent segmentation ambiguity of mention names, especially persons and locations, and, finally, how to handle the attachment of pronouns to stems. Arabic blank-delimited words are composed of zero or more prefixes, followed by a stem and zero or more suffixes. Each prefix, stem or suffix will be called a token in this discussion; any contiguous sequence of tokens can represent a mention.

For example, the word "trwmAn" (translation: "Truman") could be segmented in 3 tokens (for instance, if the word was not seen in the training data):

$$\text{trwmAn} \Rightarrow \text{t+rwm+An}$$

which introduces ambiguity, as the three tokens form really just one mention, and, in the case of the word "tmnEh", which has the segmentation

$$\text{tmnEh} \Rightarrow \text{t+mnE+h}$$

the first and third tokens should both be labeled as pronominal mentions – but, to do this, they need to be separated from the stem *mnE*.

Pragmatically, we found segmenting Arabic text to be a necessary and beneficial process due mainly to two facts:

1. some prefixes/suffixes can receive a different mention type than the stem they are glued to (for instance, in the case of pronouns);
2. keeping words together results in significant data sparseness, because of the inflected nature of the language.

[4] Each language may have a different notion of what represents a token.

[5] In the current implementation, the models use a history of 2 tags.

Feature Type	**Ar**	**Zh**	**En**
Token in window of 5	√	√	√
Morph in window of 5	√	N/A	√
POS info	√	√	√
Text chunking info	—	—	√
Capitalization/word-type	N/A	N/A	√
Prefixes/suffixes	√	N/A	√
Gazetteer info	√	√	√
Gap	—	—	√
Wordnet info	—	—	√
Segmentation	√	√	N/A
Additional systems' output	√	√	√

Table 1: Summary of features used by the 3 systems

Given these observations, we decided to "condition" the output of the system on the segmented data: the text is first segmented into tokens, and the classification is then performed on tokens. The segmentation model is similar to the one presented by Lee et al. (2003), and obtains an accuracy of about 98%.

In addition, special attention is paid to prefixes and suffixes: in order to reduce the number of spurious tokens we re-merge the prefixes or suffixes to their corresponding stem if they are not essential to the classification process. For this purpose, we collect the following statistics for each prefix/suffix ps from the ACE training data: the frequency of ps occurring as a mention by itself (M) and the frequency of ps occurring as a part of mention (P). If the ratio $\frac{M}{P}$ is below a threshold (estimated on the development data), ps is re-merged with its corresponding stem. Only few prefixes and suffixes were merged using these criteria. This is appropriate for the ACE task, since a large percentage of prefixes and suffixes are annotated as pronoun mentions[6].

In addition to the language-general features described in Section 2.3, the Arabic system implements a feature specifying for each token its original stem.

For this system, the gazetteer features are computed on words, not on tokens; the gazetteers consist of 12000 person names and 3000 location and country names, all of which have been collected by few man-hours web browsing. The system also uses features based on the output of three additional mention detection classifiers: a RRM model predicting 48 mention categories, a RRM model and a HMM model predicting 32 mention categories.

2.5 Chinese Mention Detection

In Chinese text, unlike in Indo-European languages, words neither are white-space delimited nor do they have capitalization markers. Instead of a word-based model, we build a character-based one, since word segmentation errors can lead to irrecoverable mention detection errors; Jing et al. (2003) also observe that character-based models are better performing than word-based ones for Chinese named entity recognition. Although the model is character-based, segmentation information is still useful and is integrated as an additional feature stream.

Some more information about additional resources used in building the system:

- Gazetteers include dictionaries of 10k person names, 8k location and country names, and 3k organization names, compiled from annotated corpora.
- There are four additional classifiers whose output is used as features: a RRM model which outputs 32 named categories, a RRM model identifying 49 categories, a RRM model identifying 45 mention categories, and a RRM model that classifies whether a character is an English character, a numeral or other.

2.6 English Mention Detection

The English mention detection model is similar to the system described in (Ittycheriah et al., 2003)[7].The following is a list of additional features (again, w_i is the current token):

- Shallow parsing information associated with the tokens in window of 3;
- Prefixes/suffixes of length up to 4;
- A capitalization/word-type flag (similar to the ones described by Bikel et al. (1997));
- Gazetteer information: a handful of location (55k entries) person names (30k) and organizations (5k) dictionaries;
- A combination of gazetteer, POS and capitalization information, obtained as follows: if the word is a closed-class word — select its class, else if it's in a dictionary — select that class, otherwise back-off to its capitalization information; we call this feature *gap;*
- WordNet information (the synsets and hypernyms of the two most frequent senses of the word);
- The outputs of three systems (HMM, RRM and MaxEnt) trained on a 32-category named entity data, the output of an RRM system trained on the MUC-6 data, and the output of RRM model identifying 49 categories.

3 Entity Tracking

This section introduces a novel statistical approach to entity tracking. We choose to model the *process* of forming entities from mentions, one step at a time. The process works from left to right: it starts with an initial entity consisting of the first mention of a document, and the next mention is processed by either *linking* it with one of the

[6] For some additional data, annotated with 32 named categories, mentioned later on, we use the same approach of collecting the M and P statistics, but, since named mentions are predominant and there are no pronominal mentions in that case, most suffixes and some prefixes are merged back to their original stem.

[7] The main difference between their system and ours is that they build a MaxEnt model capable of building hierarchical structures – therefore treating the problem as a parsing task – while our system treats the problem as a classification task.

existing entities, or *starting* a new entity. The process could have as output any one of the possible partitions of the mention set.[8] Two separate models are used to score the linking and starting actions, respectively.

3.1 Tracking Algorithm

Formally, let $\{m_i : 1 \leq i \leq n\}$ be n mentions in a document. Let $g : i \mapsto j$ be the map from mention index i to entity index j. For a mention index $k(1 \leq k \leq n)$, let us define

$$J_k = \{g(1), \ldots, g(k-1)\}$$

the set of indices of the partially-established entities to the left of m_k (note that $J_1 = \emptyset$), and

$$E_k = \{e_t : t \in J_k\},$$

the set of the partially-established entities.

Given that E_k has been formed to the left of the active mention m_k, m_k can take two possible actions: if $g(k) \in J_k$, then the active mention m_k is said to *link* with the entity $e_{g(k)}$; Otherwise it *start*s a new entity $e_{g(k)}$. At training time, the action is known to us, and at testing time, both hypotheses will be kept during search. Notice that a sequence of such actions corresponds uniquely to an entity outcome (or a partition of mentions). Therefore, the problem of coreference resolution is equivalent to ranking the action sequences.

In this work, a binary model $P(L = 1|E_k, m_k, A = t)$ is used to compute the link probability, where $t \in J_k$, L is 1 iff m_k links with e_t; the random variable A is the index of the partial entity to which m_k is linking. Since starting a new entity means that m_k does not link with any entities in E_k, the probability of starting a new entity, $P(L = 0|E_k, m_k)$, can be computed as

$$P(L=0|E_k,m_k) = \sum_{t \in J_k} P(L=0, A=t|E_k,m_k) = 1 - \sum_{t \in J_k} P(A=t|E_k,m_k) P(L=1|E_k,m_k,A=t) \tag{1}$$

Therefore, the probability of starting an entity can be computed using the linking probabilities $P(L = 1|E_k, m_k, A = t)$, provided that the marginal $P(A = t|E_k, m_k)$ is known. While other models are possible, in the results reported in this paper, $P(A = t|E_k, m_k)$ is approximated as:

$$P(A=t|E_k,m_k) = \begin{cases} 1 & \text{if } t = \arg\max_{i \in J_k} \\ & \quad P(L=1|E_k,m_k,A=i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

That is, the starting probability is just one minus the maximum linking probability.

Training directly the model $P(L = 1|E_k, m_k, A = i)$ is difficult since it depends on all partial entities E_k. As a first attempt of modeling the process from mentions to entities, we make the following modeling assumptions:

$$P(L=1|E_k,m_k,A=i) \approx P(L=1|e_i,m_k) \tag{3}$$

$$\approx \max_{m \in e_i} P(L=1|m,m_k). \tag{4}$$

Once the linking probability $P(L = 1|E_k, m_k, A = i)$ is available, the starting probability $P(L = 0|E_k, m_k)$ can be computed using (1) and (2). The strategy used to find the best set of entities is shown in Algorithm 3.

Algorithm 3 Coreference Decoding Algorithm

```
Input: mentions in text $M = \{m_i : 1, \ldots, n\}$
Output: a partition $E$ of the set $M$
$\mathcal{H} \leftarrow \{E_0 = \{\{m_1\}\}\}$ ; $scr(E_0) = 1$
foreach $k = 2, \ldots, n$
  $\mathcal{H}' \leftarrow \emptyset$
  foreach $E \in \mathcal{H}$
    $E' \leftarrow E \cup \{\{m_k\}\}$
    $scr(E') \leftarrow scr(E) \cdot P(L = 0|E, m_k)$
    $\mathcal{H}' \leftarrow \mathcal{H}' \cup \{E'\}$
    foreach $i \in J_k$
      $E' \leftarrow (E \setminus \{e_i\}) \cup \{e_i \cup \{m_k\}\}$
      $scr(E') \leftarrow scr(E) \cdot P(L = 1|E, m_k, A = i)$
      $\mathcal{H}' \leftarrow \mathcal{H}' \cup \{E'\}$
  $\mathcal{H} \leftarrow prune(\mathcal{H}')$
return $\arg\max_{E \in \mathcal{H}} scr(E)$
```

3.2 Entity Tracking Features

A maximum entropy model is used to implement (4). Atomic features used by the model include:

- string match – whether or not the mention strings of m and m_k are exactly match, or partially match;
- context – surrounding words or part-of-speech tags (if available) of mentions m, m_k;
- mention count – how many times a mention string appears in the document. The count is quantized;
- distance – distance between the two mentions in words and sentences. This number is also quantized;
- editing distance – quantized editing distance between the two mentions;
- mention information – spellings of the two mentions and other information (such as POS tags) if available; If a mention is a pronoun, the feature also computes gender, plurality, possessiveness and reflexiveness;
- acronym – whether or not one mention is the acronym of the other mention;
- syntactic features – whether or not the two mentions appear in apposition. This information is extracted from a parse tree, and can be computed only when a parser is available;

[8] The number of all possible partitions of a set is given by the Bell number (Bell, 1934). This number is very large even for a document with a moderate number of mentions: about 51.7 trillion for a 20-mention document. For practical reasons, the search space has to be reduced to a reasonably small set of hypotheses.

Data Set	Arabic	Chinese	English
Train	65.6k	86.5k	340.7k
Development Test	7.7k	7.2k	71k
Sep'03 Eval Test	93.5k	108.2k	60.7k

Table 2: Data statistics (number of tokens) for Arabic, Chinese and English

Another category of features is created by taking conjunction of the atomic features. For example, the model can capture how far a pronoun mention is from a named mention when the distance feature is used in conjunction with mention information feature.

As it is the case with with mention detection approach presented in Section 2, most features used here are language-independent and are instantiated from the training data, while some are language-specific, but mostly because the resources were not available for the specific language. For example, syntactic features are not used in the Arabic system due to the lack of an Arabic parser.

Simple as it seems, the mention-pair model has been shown to work well (Soon et al., 2001; Ng and Cardie, 2002). As will be shown in Section 4, the relatively knowledge-lean feature sets work fairly well in our tasks.

Although we also use a mention-pair model, our tracking algorithm differs from Soon et al. (2001), Ng and Cardie (2002) in several aspects. First, the mention-pair model is used as an approximation to the entity-mention model (3), which itself is an approximation of $P(L = 1|E_k, m_k, A = i)$. Second, instead of doing a *pick-first* (Soon et al., 2001) or *best-first* (Ng and Cardie, 2002) selection, the mention-pair linking model is used to compute a starting probability. The starting probability enables us to score the action of creating a new entity without thresholding the link probabilities. Third, this probabilistic framework allows us to search the space of all possible entities, while Soon et al. (2001), Ng and Cardie (2002) take the "best" local hypothesis.

4 Experimental Results

The data used in all experiments presented in this section is provided by the Linguistic Data Consortium and is distributed by NIST to all participants in the ACE evaluation. In the comparative experiments for the mention detection and entity tracking tasks, the training data for the English system consists of the training data from both the 2002 evaluation and the 2003 evaluation, while for Arabic and Chinese, new additions to the ACE task in 2003, consists of 80% of the provided training data. Table 2 shows the sizes of the training, development and evaluation test data for the 3 languages. The data is annotated with five types of entities: *person*, *organization*, *geo-political entity*, *location*, *facility*; each mention can be either named, nominal or pronominal, and can be either *generic* (not referring to a clearly described entity) or *specific*.

The models for all three languages are built as joint models, simultaneously predicting the type, level and genericity of a mention – basically each mention is labeled with a 3-pronged tag. To transform the problem into a classification task, we use the IOB2 classification scheme (Tjong Kim Sang and Veenstra, 1999).

4.1 The ACE Value

A gauge of the performance of an EDT system is the *ACE value*, a measure developed especially for this purpose. It estimates the normalized weighted cost of detection of *specific*-only entities in terms of *misses*, *false alarms* and *substitution errors* (entities marked *generic* are excluded from computation): any undetected entity is considered a *miss*, system-output entities with no corresponding reference entities are considered *false alarms*, and entities whose type was mis-assigned are *substitution errors*. The ACE value computes a weighted cost by applying different weights to each error, depending on the error type and target entity type (e.g. *PERSON-NAME*s are weighted a lot more heavily than *FACILITY-PRONOUN*s) (NIST, 2003a). The cumulative cost is normalized by the cost of a (hypothetical) system that outputs no entities at all – which would receive an ACE value of 0. Finally, the normalized cost is subtracted from 100.0 to obtain the ACE value; a value of 100% corresponds to perfect entity detection. A system can obtain a negative score if it proposed too many incorrect entities.

In addition, for the mention detection task, we will also present results by using the more established F-measure, computed as the harmonic mean of precision and recall – this measure gives equal importance to all entities, regardless of their type, level or genericity.

4.2 EDT Results

As described in Section 2.6, the mention detection systems make use of a large set of features. To better assert the contribution of the different types of features to the final performance, we have grouped them into 4 categories:

1. Surface features: lexical features that can be derived from investigating the words: words, morphs, prefix/suffix, capitalization/word-form flags
2. Features derived from processing the data with NLP techniques: POS tags, text chunks, word segmentation, etc.
3. Gazetteer/dictionary features
4. Features obtained by running other named-entity classifiers (with different tag sets): HMM, MaxEnt and RRM output on the 32-category, 49-category and MUC data sets.[9]

Table 3 presents the mention detection comparative results, F-measure and ACE value, on Arabic and Chinese data. The Arabic and Chinese models were built using

[9] In the English MaxEnt system, which uses 295k features, the distribution among the four classes of features is: 1:72%, 2:24%, 3:1%, 4:3%.

Feature Sets	**Arabic**		**Chinese**	
	F-measure	ACE	F-measure	ACE
1	59.7	43.1	62.6	51.1
1+2	60.8	46.0	67.1	57.7
1+2+3	63.4	51.8	68.4	67.7
1+2+3+4	68.5	53.2	68.6	74.1

Table 3: Mention detection results for the Arabic and Chinese

	Arabic	Chinese	English	
			Feb02	Sept02
ACE value	83.2	89.4	90.9	88.0

Table 4: Entity tracking results on *true* mentions

the RRM model. There are some interesting observations: first, the F-measure performance does not correlate well with an improvement in ACE value – small improvements in F-measure sometimes are paired with large relative improvements in ACE value, fact due to the different weighting of entity types. Second, the largest single improvement in ACE value is obtained by adding dictionary features, at least in this order of adding features.

For English, we investigated in more detail the way features interact. Figure 1 presents a hierarchical direct comparison between the performance of the RRM model and the MaxEnt model. We can observe that the RRM model makes better use of gazetteers, and manages to close the initial performance gap to the MaxEnt model.

Table 4 presents the results obtained by running the entity tracking algorithm on *true* mentions. It is interesting to compare the entity tracking results with inter-annotator agreements. LDC reported (NIST, 2003b) that the inter-annotator agreement (computed as ACE-values) between annotators are 73.5%, 87.3% and 87.8% for Arabic, Chinese and English, respectively. The system performance is very close to human performance on this task; this small difference in performance highlights the difficulty of the entity tracking task.

Finally, Table 5 presents the results obtained by running both mention detection followed by entity tracking on the ACE'03 evaluation data. Our submission in the evaluation performed well relative to the other participating systems (contractual obligations prevent us from elaborating further).

4.3 Discussion

The same basic model was used to perform EDT in three languages. Our approach is language-independent, in that the fundamental classification algorithm can be applied to every language and the only changes involve finding appropriate and available feature streams for each language. The entity tracking system uses even fewer language-specific features than the mention detection systems.

	Arabic	Chinese	English	
			RRM	MaxEnt
ACE value	54.5	58.8	69.7	73.4

Table 5: ACE value results for the three languages on ACE'03 evaluation data.

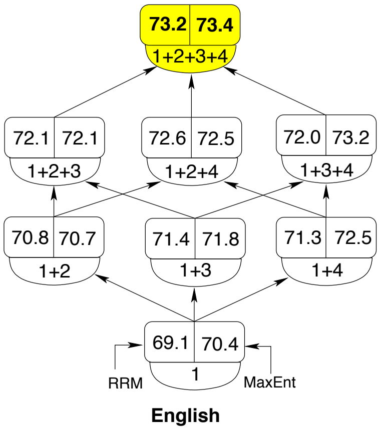

Figure 1: Performance of the English mention detection system on different sets of features (uniformly penalized F-measure), September'02 data. The lower part of each box describes the particular combination of feature types; the arrows show a inclusion relationship between the feature sets.

One limitation apparent in our mention detection system is that it does not model explicitly the genericity of a mention. Deciding whether a mention refers to a specific entity or a generic entity requires knowledge of substantially wider context than the window of 5 tokens we currently use in our mention detection systems. One way we plan to improve performance for such cases is to separate the task into two parts: one in which the mention type and level are predicted, followed by a genericity-predicting model which uses long-range features, such as sentence or document level features.

Our entity tracking system currently cannot resolve the coreference of pronouns very accurately. Although this is weighted lightly in ACE evaluation, good anaphora resolution can be very useful in many applications and we will continue exploring this task in the future.

The Arabic and Chinese EDT tasks were included in the ACE evaluation for the first time in 2003. Unlike the English case, the systems had access to only a small amount of training data (60k words for Arabic and 90k characters for Chinese, in contrast with 340k words for English), which made it difficult to train statistical models with large number of feature types. Future ACE evaluations will shed light on whether this lower performance, shown in Table 3, is due to lack of training data or to specific language-specific ambiguity.

The final observation we want to make is that the systems were not directly optimized for the ACE value, and there is no obvious way to do so. As Table 3 shows, the F-measure and ACE value do not correlate well: systems trained to optimize the former might not end up optimizing the latter. It is an open research question whether a system can be directly optimized for the ACE value.

5 Conclusion

This paper presents a language-independent framework for the entity detection and tracking task, which is shown to obtain top-tier performance on three radically different languages: Arabic, Chinese and English. The task is separated into two sub-tasks: a mention detection part, which is modeled through a named entity-like approach, and an entity tracking part, for a which a novel modeling approach is proposed.

This statistical framework is general and can incorporate heterogeneous feature types — the models were built using a wide array of lexical, syntactic and semantic features extracted from texts, and further enhanced by adding the output of pre-existing semantic classifiers as feature streams; additional feature types help improve the performance significantly, especially in terms of ACE value. The experimental results show that the systems perform remarkably well, for both well investigated languages, such as English, and for the relatively new additions Arabic and Chinese.

6 Acknowledgements

We would like to thank Dr. Tong Zhang for providing us with the RRM toolkit.

This work was partially supported by the Defense Advanced Research Projects Agency and monitored by SPAWAR under contract No. N66001-99-2-8916. The views and findings contained in this material are those of the authors and do not necessarily reflect the position of policy of the U.S. government and no official endorsement should be inferred.

References

J. Aberdeen, D. Day, L. Hirschman, P. Robinson, and M. Vilain. 1995. Mitre: Description of the Alembic system used for MUC-6. In *Proceedings of MUC-6*, pages 141–155.

E. T. Bell. 1934. Exponential numbers. *American Math. Monthly*, 41:411–419.

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: a high-performance learning namefinder. In *Proceedings of ANLP-97*, pages 194–201.

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition.

A. Ittycheriah, L. Lita, N. Kambhatla, N. Nicolov, S. Roukos, and M. Stys. 2003. Identifying and tracking entity mentions in a maximum entropy framework. In *HLT-NAACL 2003: Short Papers*, May 27 - June 1.

H. Jing, R. Florian, X. Luo, T. Zhang, and A. Ittycheriah. 2003. HowtogetaChineseName(Entity): Segmentation and combination issues. In *Proceedings of EMNLP'03*, pages 200–207.

Y.-S. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan. 2003. Language model based Arabic word segmentation. In *Proceedings of the ACL'03*, pages 399–406.

A. Mikheev, M. Moens, and C. Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of EACL'99*.

S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwarz, R. Stone, and R. Weischedel. 1998. Bbn: Description of the SIFT system as used for MUC-7. In *MUC-7*.

G. A. Miller. 1995. WordNet: A lexical database. *Communications of the ACM*, 38(11).

V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the ACL'02*, pages 104–111.

NIST. 2003a. The ACE evaluation plan. www.nist.gov/speech/tests/ace/index.htm.

NIST. 2003b. Proceedings of ACE'03. Booklet, Alexandria, VA, September.

L. Ramshaw and M. Marcus. 1994. Exploring the statistical derivation of transformational rule sequences for part-of-speech tagging. In *Proceedings of the ACL Workshop on Combining Symbolic and Statistical Approaches to Language*, pages 128–135.

L. Ramshaw and M. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of WVLC'95*, pages 82–94.

W. M. Soon, H. T. Ng, and C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

E. F. Tjong Kim Sang and J. Veenstra. 1999. Representing text chunks. In *Proceedings of EACL'99*.

E. F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158.

R. J. Williams and J. Peng. 1990. An efficient gradient–based algorithm for on–line training of recurrent neural networks trajectories. *Neural Computation*, 2(4):490–501.

T. Zhang, F. Damerau, and D. E. Johnson. 2002. Text chunking based on a generalization of Winnow. *Journal of Machine Learning Research*, 2:615–637.